International Journal of Engineering Research-Online
A Peer Reviewed International Journal
Articles available online http://www.ijoer.in

Vol.3., Issue.1, 2015

RESEARCH ARTICLE

ISSN: 2321-7758

FRAME LEVEL PIPELINE MOTION ESTIMATION ARRAY PROCESSOR

S.IMMANUEL PRABAHARAN[1], J.REEGAN[2], M.SELVA KUMAR[3], J.ABISHA NIRAIMATHI[3]
[1,2,3]Assistant Professor, [3]PG Scholar,
Chandy College of Engineering, Tuticorin, India

Article Received: 15/02/2015 | Article Revised on:18/02/2015 | Article Accepted on:21/02/2015

S.IMMANUEL PRABAHARAN

ABSTRACT

Given the critical role of motion estimation (ME) in a video coder, testing such a module is of priority concern. While focusing on the testing of ME in a video coding system, this work presents an error detection and data recovery (EDDR) design, based on the residue-and-quotient (RQ) code, to embed into ME for video coding testing applications. An error in processing elements (PEs), i.e. key components of a ME, can be detected and recovered effectively by using the proposed EDDR design. Experimental results indicate that the proposed EDDR design for ME testing can detect errors and recover data with an acceptable area overhead and timing penalty. Importantly, the proposed EDDR design performs satisfactorily in terms of throughput and reliability for ME testing applications.

Key Words—Area overhead, data recovery, error detection, motion estimation, reliability, residue-and-quotient (RQ) code.

INTRODUCTION

Advances in semiconductors, digital signal processing, and communication technologies have made multimedia applications more flexible and reliable. A good example is the H.264 video standard, also known as MPEG-4 Part 10 Advanced Video Coding, which is widely regarded as the next generation video compression standard [1], [2]. Video compression is necessary in a wide range of applications to reduce the total data amount required for transmitting or storing video data. Among the coding systems, a ME is of priority concern in exploiting the temporal redundancy between successive frames, yet also the most time consuming aspect of coding. Additionally, while performing up to 60%–90% of the computations encountered in the entire coding system, a ME is widely regarded as the most computationally intensive of a video coding system [3]. AME generally consists of PEs with a size of 4 4. However, accelerating the computation speed depends on a large PE array, especially in high-resolution devices with a large search range such as HDTV [4]. Additionally, the visual quality and peak signal-to-noise ratio (PSNR) at a given bit rate are influenced if an error occurred in ME process. A testable design is thus increasingly important to ensure the reliability of numerous PEs in a ME. Moreover, although the advance of VLSI technologies facilitate the integration of a large number of PEs of aME into a chip, the logic-per-pin ratio is subsequently increased, thus decreasing significantly the efficiency of logic testing on the chip. As a commercial chip, it is absolutely necessary for the ME to introduce design for testability (DFT) [5]–[7]. DFT focuses on increasing the ease of device testing, thus guaranteeing high reliability of a system. DFT methods rely on reconfiguration of a circuit under test (CUT) to improve testability. While DFT approaches enhance the testability of circuits, advances in sub-micron technology and resulting increases in the complexity of electronic circuits and systems have meant that built-in self-test (BIST) schemes have rapidly become necessary in the digital world. BIST for the ME does not expensive test equipment, ultimately lowering test costs [8]–[10]. Moreover, BIST can generate test simulations and analyse test responses without outside support, subsequently streamlining the testing and diagnosis of digital systems. However, increasingly complex density of circuitry requires that the built-in testing approach not only detect faults but also specify their locations for error correcting. Thus, extended schemes of BIST referred to as built-in self-diagnosis [11] and built-in self-correction [12]–[14] have been developed recently.

II. RQ CODE GENERATION

Coding approaches such as parity code, Berger code, and residue code have been considered for design applications to detect circuit errors [17], [18]. Residue code is generally separable arithmetic codes by estimating a residue for data and appending it to data. Error detection logic for operations is typically derived by a separate residue code, making the detection logic is simple and easily implemented. For instance, assume that denotes an integer, and represent data words, and refers to the modulus. A separate residue code of interest is one in which is coded as a pair . Notably, is the residue of modulo . Error detection logic for operations is typically derived using a separate residue code such that detection logic is simply and easily implemented. However, only a bit error can be detected based on the residue code. Additionally, an error cannot be recovered effectively by using the residue codes. Therefore, this work presents a quotient code, which is derived from the residue code, to assist the residue code in detecting multiple errors and recovering errors. The mathematical model of RQ code is simply described as follows. Assume that binary data is expressed as

$$X=\{b_{n-1}\, b_{n-2} \,.......b_2\, b_1\; b_0\}=\sum_{j=0}^{n-1} b_j 2^j$$

According to the above RQ code expression, the corresponding circuit design of the RQCG can be realized. In order to simplify the complexity of circuit design, the implementation of the module is generally dependent on the addition operation. Additionally, based on the concept of residue code, the following definitions shown can be applied to generate the RQ code for circuit design.

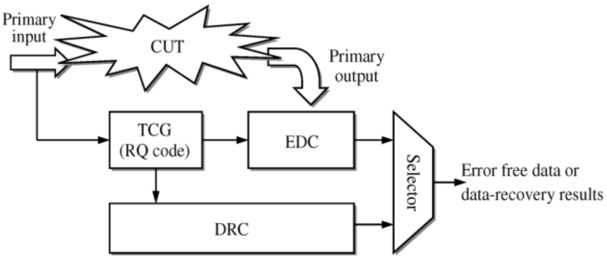


Fig. 1. Conceptual view of the proposed EDDR architecture.

III. PROPOSED EDDR ARCHITECTURE DESIGN

Fig. 1 shows the conceptual view of the proposed DDR scheme, which comprises two major circuit designs, i.e. error detection circuit (EDC) and data recovery circuit (DRC), to detect errors and recover the corresponding data in a specific CUT. The test code generator (TCG) in Fig. 1 utilizes the concepts of RQ code to generate the corresponding test codes for error detection and data recovery. In other words, the test codes from TCG and the primary output from CUT are delivered to EDC to determine whether the CUT has errors. DRC is in charge of recovering data from TCG. Additionally, a selector is enabled to export error-free data or data-recovery results. Importantly, an array-based computing structure, such as ME, discrete cosine transform (DCT), iterative logic array (ILA), and finite impulse filter (FIR), is feasible for the proposed EDDR scheme to detect errors and recover the corresponding data.

A. Fault Model

The PEs are essential building blocks and are connected regularly to construct a ME. Generally, PEs are surrounded by sets of ADDs and accumulators that determine how data flows through them. PEs can thus be considered the class of circuits called ILAs, whose testing assignment can be easily achieved by using the fault model, cell fault model (CFM) [21]. Using CFM has received considerable interest due to accelerated growth in the use of high-level synthesis, as well as the parallel increase in complexity and density of integration circuits (ICs). Using CFM makes the tests independent of the adopted synthesis tool and vendor library. Arithmetic modules, like ADDs (the primary element in a PE), due to their regularity, are designed in an extremely dense configuration.

B. TCG Design

According to Fig. 2, TCG is an important component of the proposed EDDR architecture. Notably, TCG design is based on the ability of the RQCG circuit to generate corresponding test codes in order to detect errors and recover data. The specific in Fig. 2 estimates the absolute difference between the Cur_pixel of the search area and the Ref_pixel of the current macro block.

C. EDDR Processes

Fig. 2 clearly indicates that the operations of error detection in a specific is achieved by using EDC, which is utilized to compare the outputs between TCG and in order to determine whether errors have occurred. If the values of and/or , then the errors in a specific can be detected. The EDC output is then used to generate a 0/1 signal to indicate that the tested is error-free/cranky.

	0	1	2	3
0	128	128	64	255
1	128	64	255	64
2	64	255	64	128
3	255	64	128	128

Cur_pixel

	0	1	2	3
0	1	1	2	3
1	1	2	3	4
2	2	3	4	5
3	3	4	5	5

Ref_pixel

Fig. 2. Example of pixel values.

D. Numerical Example

A numerical example of the 16 pixels for a 4*4 macro block in a specific of a ME is described as follows. Fig. 5 presents an example of pixel values of the Cur_pixel and Ref_pixel. Based on (7), the SAD value of the 4*4 macroblock is

$$\begin{aligned}\mathrm{SAD} &= \sum_{i=0}^{3}\sum_{j=0}^{3}|X_{ij}-Y_{ij}|\\ &= |X_{00}-Y_{00}|+|X_{01}-Y_{01}|+\ldots+|X_{33}-Y_{33}|\\ &= (128-1)+(128-1)+\ldots+(128-5)\\ &= 2124.\end{aligned}$$

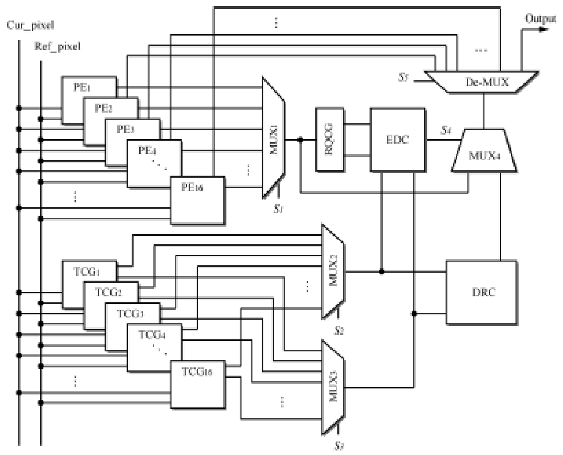


Fig. 3. Proposed EDDR architecture design for a ME.

E. Overall Test Strategy

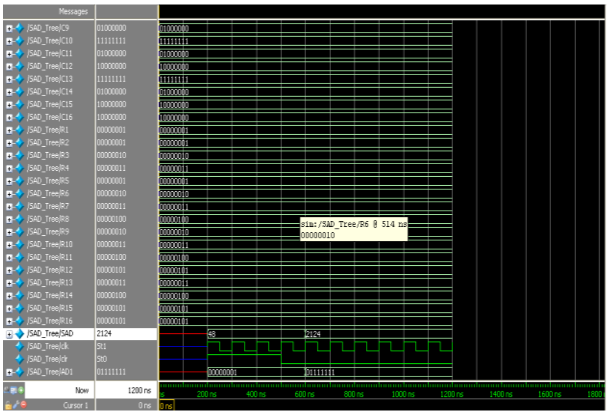
By extending the testing processes of a specific in Fig. 2, Fig. 6 illustrates the overall EDDR architecture design of a ME. First, the input data of Cur_pixel and Ref_pixel are sent simultaneously to PEs and TCGs in order to estimate the SAD values and generate the test RQ code and . Second, the SAD value from the tested object , which is selected by , is then sent to the RQCG circuit in order to generate and codes. Meanwhile, the corresponding test codes and from a specific are selected simultaneously by MUXs 2 and 3, respectively. Third, the RQ code from and RQCG circuits are compared in EDC to determine whether the tested object have errors. The tested object is error-free if and only if and . Additionally, DRC is used to recover data encoded by , i.e. the appropriate and codes from are selected by MUXs 2 and 3, respectively, to recover data. Fourth, the error-free data or data recovery results are selected by . Notably, control signal is generated from EDC, indicating that the comparison result is error-free. Finally, the error-free data or the data-recovery result from the tested object is passed to a De-MUX, which is used to test the next specific ; otherwise, the final result is exported.

IV. RESULTS AND DISCUSSION

Extensive verification of the circuit design is performed using the VHDL and then synthesized by the Synopsys Design Compiler with TSMC 0.18- m 1P6M CMOS technology to demonstrate the feasibility of the proposed EDDR architecture design for ME testing applications.

The TCG component plays a major role in the proposed EDDR architecture to detect errors and recover data.

Additionally, the number of TCGs significantly influences the circuit performance in terms of area overhead and throughput. Figs. 7 and 8 illustrate the relations between the number of TCGs, area overhead and throughput. The area overhead is less than 2% if only one TCG is used to execute; however, at this time, the throughput is extremely small. Notably, the throughput of a ME without embedding the proposed EDDR architecture is about 25 800 kMB/s. Fig. 8 clearly indicates that the throughput is around 25 000 kMB/s, if the proposed EDDR architecture with 16 TCGs is embedded into a ME for testing. Thus, to maintain the same throughput as much as possible, 16 TCGs must be adopted in the proposed EDDR architecture for a ME testing applications. Although the area overhead is increased if 16 TCGs used (see Fig. 7), the area overhead is only about 5.13%, i.e. an acceptable design for circuit testing.



```
Messages
/SAD_Tree/C9     01000000
/SAD_Tree/C10    11111111
/SAD_Tree/C11    01000000
/SAD_Tree/C12    10000000
/SAD_Tree/C13    11111111
/SAD_Tree/C14    01000000
/SAD_Tree/C15    10000000
/SAD_Tree/C16    10000000
/SAD_Tree/R1     00000001
/SAD_Tree/R2     00000001
/SAD_Tree/R3     00000010
/SAD_Tree/R4     00000011
/SAD_Tree/R5     00000001
/SAD_Tree/R6     00000010
/SAD_Tree/R7     00000011
/SAD_Tree/R8     00000100
/SAD_Tree/R9     00000010
/SAD_Tree/R10    00000011
/SAD_Tree/R11    00000100
/SAD_Tree/R12    00000101
/SAD_Tree/R13    00000011
/SAD_Tree/R14    00000100
/SAD_Tree/R15    00000101
/SAD_Tree/R16    00000101
/SAD_Tree/SAD    2124
/SAD_Tree/clk    St1
/SAD_Tree/clr    St0
/SAD_Tree/AD1    01111111
Now              1200 ns
Cursor 1         0 ns
```

sim:/SAD_Tree/R6 @ 514 ns
00000010

Fig 4. simulation output for SAD tree

This work also addresses reliability-related issues to demonstrate the feasibility of the proposed EDDR architecture. Reliability is the probability that a component or a system performs its required function under different operating conditions encountered for a certain time period. The constant failure rate reliability model.

CONCLUSION

This work presents an EDDR architecture for detecting the errors and recovering the data of PEs in a ME. Based on the RQ code, a RQCG-based TCG design is developed to generate the corresponding test codes to detect errors and recover data. The proposed EDDR architecture is also implemented by using VHDL and synthesized by the Synopsys Design Compiler with TSMC 0.18- m1P6MCMOStechnology. Experimental results indicate that that the proposed EDDR architecture can effectively detect errors and recover data in PEs of a ME with reasonable area overhead and only a slight time penalty.

REFERENCES

[1]. *Advanced Video Coding for Generic Audiovisual Services*, ISO/IEC 14496-10:2005 (E), Mar. 2005, ITU-T Rec. H.264 (E).

[2]. *Information Technology-Coding of Audio-Visual Objects—Part 2: Visual*, ISO/IEC 14 496-2, 1999.

[3]. Y. W. Huang, B. Y. Hsieh, S. Y. Chien, S. Y. Ma, and L. G. Chen, "Analysis and complexity reduction of multiple reference frames motion estimation in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 4, pp. 507–522, Apr. 2006.

[4]. C. Y. Chen, S. Y. Chien, Y. W. Huang, T. C. Chen, T. C. Wang, and L. G. Chen, "Analysis and architecture design of variable block-size motion estimation for H.264/AVC," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 53, no. 3, pp. 578–593, Mar. 2006.

[5]. T. H. Wu, Y. L. Tsai, and S. J. Chang, "An efficient design-for-testability scheme for motion estimation in H.264/AVC," in *Proc. Int. Symp. VLSI Design, Autom. Test*, Apr. 2007, pp. 1–4.

[6]. M. Y. Dong, S. H. Yang, and S. K. Lu, "Design-for-testability techniques for motion estimation computing arrays," in *Proc. Int. Conf. Commun., Circuits Syst.*, May 2008, pp. 1188–1191.

[7]. Y. S. Huang, C. J. Yang, and C. L. Hsu, "C-testable motion estimation design for video coding systems," *J. Electron. Sci. Technol.*, vol. 7, no. 4, pp. 370–374, Dec. 2009.

[8]. D. Li, M. Hu, and O. A. Mohamed, "Built-in self-test design of motion estimation computing array," in *Proc. IEEE Northeast Workshop Circuits Syst.*, Jun. 2004, pp. 349–352.

[9]. Y. S. Huang, C. K. Chen, and C. L. Hsu, "Efficient built-in self-test for video coding cores: A case study on motion estimation computing array," in *Proc. IEEE Asia Pacific Conf. Circuit Syst.*, Dec. 2008, pp. 1751–1754.

[10]. W. Y Liu, J. Y. Huang, J. H. Hong, and S. K. Lu, “Testable design and BIST techniques for systolic motion estimators in the transform domain,” in *Proc. IEEE Int. Conf. Circuits Syst.*, Apr. 2009, pp. 1–4.

[11]. J. M. Portal, H. Aziza, and D. Nee, “EEPROM memory: Threshold voltage built in self diagnosis,” in *Proc. Int. Test Conf.*, Sep. 2003, pp. 23–28.

[12]. J. F. Lin, J. C. Yeh, R. F. Hung, and C. W. Wu, “A built-in self-repair design for RAMs with 2-D redundancy,” *IEEE Trans. Vary Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 6, pp. 742–745, Jun. 2005.

[13]. C. L. Hsu, C. H. Cheng, and Y. Liu, “Built-in self-detection/correctionarchitecture for motion estimation computing arrays,” *IEEE Trans.Vary Large Scale Integr. (VLSI) Systs.*, vol. 18, no. 2, pp. 319–324, Feb.2010.

[14]. C. H. Cheng, Y. Liu, and C. L. Hsu, “Low-cost BISDC design formotion estimation computing array,” in *Proc. IEEE Circuits Syst. Int.Conf.*, 2009, pp. 1–4.

[15]. S. Bayat-Sarmadi and M. A. Hasan, “On concurrent detection of errorsin polynomial basis multiplication,” *IEEE Trans. Vary Large Scale Integr.(VLSI) Systs.*, vol. 15, no. 4, pp. 413–426, Apr. 2007.

[16]. C. W. Chiou, C. C. Chang, C. Y. Lee, T. W. Hou, and J. M. Lin,“Concurrent error detection andcorrection in Gaussian normal basismultiplier over GF □_ __,” *IEEE Trans. Comput.*, vol. 58, no. 6, pp.851–857, Jun. 2009.

[17]. L. Breveglieri, P. Maistri, and I. Koren, “A note on error detection in anRSA architecture by means ofresidue codes,” in *Proc. IEEE Int. Symp. On-Line Testing*, Jul. 2006, 176 177.

[18]. S. J. Piestrak, D. Bakalis, and X. Kavousianos, “On the design of selftestingcheckers for modified Bergercodes,” in *Proc. IEEE Int. WorkshopOn-Line Testing*, Jul. 2001, pp. 153–157.

[19]. S. Surin and Y. H. Hu, “Frame-level pipeline motion estimation arrayprocessor,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 2,pp. 248–251, Feb. 2001.

[20]. D. K. Park, H. M. Cho, S. B. Cho, and J. H. Lee, “A fast motion estimationalgorithm for SAD optimization in sub-pixel,” in *Proc. Int. Symp. Integr. Circuits*, Sep. 2007, pp. 528–531